

B.D.U.C.

BETA DISK USERS CLUB

BETA DISK NEWSLETTER NO. 5

Hello again. First, some news about a new product for the Beta. It's a little bit different than anything previously available. A German software house called Individual Software has produced a replacement EPROM for the Beta, it's called VISION and it can be used with 3.xx and 4.xx Beta interfaces.

The EPROM contains the Beta DOS software as normal but additional code has been implemented to produce a program that provides a new operating environment for Beta DOS, it's basically a desktop package and provides a means of driving Beta DOS from pull down menus and pointers. The pointer may be controlled via the keyboard, Kempston mouse or keyboard mouse (a device produced by Individual, which simulates a joystick and is plugged in a joystick interface).

Most functions can be carried out with single keypresses from menus. The technical information states the package is capable of supporting upto four drives and is able to drive printers. The price of the package as a 16K replacement EPROM is £15 plus £2 P & P. Individual also produce a range of other Beta related software packages. The newsletter will hopefully have a review in the next issue.

More information can be obtained direct from Individual Software, Volker Marohn, Am Beilsteck 30, 4600 Dortmund 50, West Germany or see the enclosed information sheet.

More about new products. Myrmidon Software has just released an updated version of 'The Last Word' word processor package, to be marketed by Trojan Products. Nick Buckingham who runs Myrmidon tells me it's a much improved version with many new and revamped features including a re-written Beta DOS handling routine. Also, the previously separate extension software has been integrated into the main package. The specification certainly looks impressive. Myrmidon originally wrote TLW for the now defunkt Saga Systems who cclapsed several months ago. Myrmidon is providing support for TLW users.

Another liquidation was announced recently that of Kempston (of joystick interface fame). Kempston also produced a disk interface known as the KDOS but support for this interface was never strong, fewer software houses supported the KDOS than Beta DOS, although versions of Art Studio using KDOS were produced.

SPLITTER PROGRAM MODIFICATION FOR 5.XX TRDOS. BY M.J.SMITH BDOC.

Here are the modifications for using Bernhard Lutz' splitter program as featured in the last issue. The instructions for creating the main program should be followed but before the program is saved the following modifications to the listing should be included.

1. Load your splitter basic from the last issue add/edit lines as follows.

```

10 LET DOS = VAL "15619"
300 RANDOMIZE USR VAL "15616"
8999 STOP: REM LOADER 27000,20
9000 DATA 0,0,0,33,0,0,22,0,30,0,6,1,0,0,0,14,5,205,19,61,201
    
```

2. Delete lines 9010-9030, these are no longer required;

3. Save as described.

VERSION 4.XX DISASSEMBLY. (27000=6978H)

```

6978 CD6B3C CALL 3C6B ;switch DOS on (15467)
697B 210000 LD HL,0000 ;buffer address
697E 1600 LD D,00 ;track
6980 1E00 LD E,00 ;sector number
6982 3E01 LD A,01 ;length
6984 0601 LD B,01 ;number of sectors
6986 0E00 LD C,00 ;
6988 F5 PUSH AF ;save value
6989 C5 PUSH BC ;save value
698A D5 PUSH DE ;save value
698B CDD62E CALL 2ED6 ;read disk (11900)
698E D1 POP DE ;return value
698F C1 POP BC ;return value
6990 1C INC E ;sector counter+1
6991 3E10 LD A,10 ;load A reg,sectors/track.
6993 BB CP E ;if sector=16
6994 CCA169 CALL Z,69A1 ;jump to track counter
6997 F1 POP AF ;return value
6998 3D DEC A ;
6999 2802 JR Z,699D ;all done switch DOS off
699B 1BE7 JR 6984 ;do another sector
699D CD7C3C CALL 3C7C ;switch DOS off (15484)
69A0 C9 RETURN ;end
69A1 1E00 LD E,00 ;zero sector counter
69A3 14 INC D ;track+1
69A4 C9 RET ;return
    
```

The modification for 5.XX DOS is quite simple. In the version 4.XX disassembly above the program counts the tracks read and increments the sector counter when 16 tracks have been processed. For version 5.XX the DOS takes care of this process and the coding is now simplified. The DOS routine is called with the total number of sectors to read in the B register.

VERSION 5.XX DISASSEMBLY. (27000=6978H)

```

6978 00      NOP                ;make space
6979 00      NOP
6980 00      NOP
6981 210000 LD  HL,0000        ;buffer address
6984 1600    LD  D,00          ;load D reg track number
6986 1E00    LD  E,00          ;load E reg sector number
6988 060100 LD  B,00          ;number of sectors
698B 00      NOP                ;make space
698C 00      NOP
698D 0E05    LD  C,05          ;routine 5, read data
698F CD133D CALL 3D13          ;do it
6992 C9      RET                ;finish

```

Remember the magic file must be located on track 1 sector 0 for this program to operate correctly this is achieved by saving the snapshot on a clean disk.

Another enhancement would be to have the program automatically save all four combinations of interrupt mode and interrupts enabled or disabled. It would then be much simpler to find the split version that works. The program could use suffix A to D thus.

```

File A IM1, interrupts enabled.
File B IM1, interrupts disabled.
File C IM2, interrupts enabled.
File D IM2, interrupts disabled.

```

This feature could be added as another subroutine, but memory space for the basic part is limited so other features like the menu section could be overwritten. In this version there would not be any requirement to select options from a menu since the program would be automatic. Anybody willing to do the mods?

FILE UNDELETE. TRDOS 5.XX. BY M. J. SMITH. BDUC.

This short program enables files to be recovered from a disk if they have been previously ERASED from the disk catalogue, however files cannot be recovered if a MOVE has been executed since this operation repacks and totally deletes the program from the disk.

The code to call routine number 5, read data (see last issue) is loaded by line 30 from the data in line 40.

The data for the files is contained on track 0 sector 0 and the catalogue is 16 sectors long (4096 bytes). This is read by the code and the data is stored in buffer address 32000. The loop in lines 120 to 160 searches for the deleted file marker in this case 1 and replaces it with another marker that the DOS recognises as a legal name for display. The whole buffer is now written back to track 0 sector 0 using routine number 6.

The recovered file will now begin with the character ! and may be loaded normally. The program checks the disk catalogue for the erased file marker and returns with the message "No Files" if an attempt is made to undelete files from a disk with no files tagged with the erased marker 1.

The recovered file may be renamed and the MOVE function may be used as normal. The code is similar to that for the splitter program above. The information for the routine is POKED from within the program.

There may be a problem with undeleting files if the same characters appear in a filename after character 2. Also the program will not undelete the last deleted file in the directory, since TRDOS does not rewrite track zero, it just forgets that the last file was ever created and uses the file space when accessing the disk for saving.

Listing 1.

```
10 CLEAR 29999
20 LET F=0: LET P=0: LET N$=""
30 CLS : PRINT AT 10,0;"FILE UNDELETE""READING CATALOGUE...P
   LEASE WAIT"
40 RESTORE : FOR A=30000 TO 30020: READ D: POKE A,D: NEXT A
50 DATA 0,0,0,33,0,0,22,0,30,0,6,1,0,0,0,14,5,205,19,61,201
60 LET B2=32000: LET T=0: LET S=0: LET L=16
70 LET HI=INT (B2/256): LET LO=B2-HI*256
80 POKE 30004,LO: POKE 30005,HI
90 POKE 30007,T
100 POKE 30009,S
110 POKE 30011,L
120 RANDOMIZE USR 30000
130 IF P=1 THEN POKE (B2+244),0: LET P=2: GO TO 210
140 FOR X=B2 TO B2+(L*256) STEP 16
150 FOR I=0 TO 7
160 LET N$=N$+CHR$ (PEEK (X+I))
170 IF (PEEK (X))=1 THEN POKE (X),33: LET F=1
180 NEXT X
190 IF F=0 THEN GO TO 260
200 PRINT "FILES UNDELETED""WRITING CATALOGUE...PLEASE WAIT"
```

```

200 PRINT "FILES UNDELETED""WRITING CATALOGUE...PLEASE WAIT"
210 POKE 30016,6
220 RANDOMIZE USR 30000
230 IF P=1 OR P=2 THEN GO TO 270
240 LET BZ=32000: LET T=0: LET S=8: LET L=1: POKE 30016,5: LET
    P=1
250 GO TO 70
260 PRINT "NO FILES"

```

DOS ROUTINES - PART 2 EXAMPLES. BY HENDRICK BROOTHAERS.

This is part two of this article, it contains three examples of how to use the DOS routines from machine code for 4.XX DOS.

1. The following example send a catalogue to the screen.

EXAMPLE ONE

```

CALL 3C06 ;switch DOS on
PUSH HL ;put the DOS off return address on the stack
LD A,02 ;select stream 2 (screen)
LD C,07 ;set for DOS routine 7 (catalogue)
CALL 3BFD ;call DOS to execute routine 7
RET ;return via DOS off address

```

2. The next examples is a combination of three things.

This code must be entered with the DE register pair pointing to a memory area that holds a file name and type. First the filename is moved to the DOS variables using routine 19, now the disk catalogue is searched to see if the file exists on the diskette using routine 10, if the file is not in the catalogue then address 5D0F contains FF, if the file exists, address 5D0F contains the number of the file in the catalogue.

Finally the information for the file is moved from track zero to the DOS variables. The data moved has the file length, sector, track and is moved by routine 8

EXAMPLE TWO

```

CALL 3C06 ;switch DOS on
PUSH HL ;push return address
EX DE,HL ;make HL point to the file name
LD C,13 ;set for DOS routine 19 (13H)
CALL 3BFD ;CALL routine 19, move filename to DOS vars
LD A,43 ;character "C" code 67 (43H)

```

```

LD      (5CE5),A ;set file type in DOS variable for file type
LD      C,0A     ;set for DOS routine 10 (0AH)
CALL    3BFD     ;CALL routine 10, search file in disk catalogue
LD      A,(5D0F) ;get result in A register
CP      FF      ;check if file exists
RET     Z        ;return if file does not exist
LD      C,08     ;set for DOS routine 8
CALL    3BFD     ;CALL routine 8,move file info to DOS vars
RET     ;return via DOS off address

```

3. This example shows how routines may be called from within other routines.

Lines 1 to 10 are a routine to LOAD a file from disk, when the routine is entered at line 1, the file is loaded to memory starting at A000. When the routine is entered at line 2, the file is loaded in memory starting at the address held in the DE register on entry.

EXAMPLE THREE

```

1 LD     DE,A000  ;set DE to buffer address 40960 (A000)
2 CALL   3C06     ;switch DOS on
3 PUSH  HL       ;save return address
4 CALL   32       ;CALL routine to select drive
5 EX     DE,HL   ;get buffer address in HL register pair
6 XOR   A        ;clear the A register
7 LD     (5D10),A ;set 5D10 to 0 (see routine 14 information)
8 LD     A,1     ;A register=1 (load to address in HL)
9 LD     C,14    ;set for DOS routine 14 to load the file
10 JR   24       ;go execute routine 14 and return via DOS off.

```

Lines 20 to 25 catalogue the disk to the screen.

```

20 CALL  3C06     ;switch DOS on
21 PUSH  HL       ;save return
22 LD    A,02     ;select stream 2=screen
23 LD    C,07     ;set for DOS routine 7=catalogue
24 CALL  3BFD     ;execute routine in C register
25 RET                ;return

```

Lines 30 to 42 selects the drive from the number in the memory location A123 number is in memory location A123, then calls the catalogue routine.

```

30 CALL  3C06     ;switch DOS on
31 PUSH  HL       ;save return address
32 PUSH  DE       ;save value
33 PUSH  BC       ;save value

```

```

33 PUSH BC ;save value
34 PUSH AF ;save value
35 LD A,(A123) ;load A register with drive number
36 LD C,01 ;set for DOS routine 1=select drive
37 CALL 24 ;go select drive (via the call in line 24)
38 CALL 22 ;do a CAT
39 POP AF ;get value back
40 POP BC ;get value back
41 POP DE ;get value back
42 RET ;return

```

This is the last part of the Machine code and Beta DOS article.
Thanks again to Hendrick Broothaers in Belgium.

FLOPPY DISK DRIVE- USER LINKS. BY M.J.SMITH. BDOC.

The Beta interface is capable of supporting multiple drives which are handled automatically by the interface firmware and floppy disk control chip. If multiple drives are used some knowledge of floppy drive link options is required, especially if the user wishes to upgrade from a single drive.

Here is an explanation of the links for most types of drive, these are based around the Mitsubishi MF503A, 5.25" double sided drive, (similar to the drives supplied by TR), but are also applicable to most drives of Japanese origin.

1:Drive Select.

These links determine the physical identity of the drive, for the first drive, logical unit 0 or drive A link DS0 is made. For the second drive logical unit 1 or drive B, DS1 is made.

Drive 1	DS0 - IN	Drive 2	DS0 - OUT
	DS1 - OUT		DS1 - IN

BETA CONFIG: Drive A - DS0 IN Drive B - DS1 IN

If MX is in place the drive will react to any drive select command. Hence would normally be out in most applications.

BETA CONFIG: MX - OUT

2:Motor Commands.

These links determine various functions for the drive as follows:

Motor starts by motor on command.	MM - IN	MS - OUT
Motor starts by drive select.	MM - OUT	MS - OUT
Motor starts by motor on or drive select.	MM - OUT	MS - OUT
Motor starts by IN USE latched by drive select. pin 4.	MM - IN	MS - IN
	IU - IN	

BETA CONFIG: MM - IN MS - OUT

3: Status Commands.

Standard Ready.	2S - OUT	DC - OUT
Hold Ready.	2S - IN	DC - OUT
Disk Change. (Reset by drive select)	2S - IN	DC - IN
Termination resistor. Used in last physical drive.	TD - last drive.	

BETA CONFIG: 2S - OUT DC - OUT TD - IN (last drive)

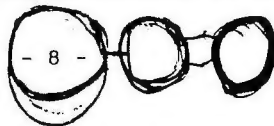
4: Indicator.

LED will not light.	IU - OUT	IS - IN
LED will light with drive select.	IU - OUT	IS - OUT
LED will light with IN USE signal.	IU - IN	IS - OUT
LED will light with the logical sum of the drive select and IN USE signals.	IU - IN	IS - OUT
	IL - OUT	

BETA CONFIG: IU - OUT IS - OUT IL - OUT

Please note this information is intended as a guide only, you should consult the manufacturer's specification sheets to verify correct link settings.

B D U C, 2, DOWNHAM AVENUE, RAWTENSTALL, ROSSENDALE,
LANCASHIRE, BB4 8JY. ENGLAND. PHONE: 010 44 706 218354



Ellet -